Using Software Metrics as Demonstrators of Design Changes in Iterative Software Processes

Mohammad Alshayeb
Computer Science Department
University of Alabama in Huntsville
April 9, 2002

1

Agenda

- Research Objectives
- Research Hypotheses
- Metrics Suites
- Previous Software Metrics Studies
- Software Processes
- Research Data
- Research Approach
- Research Significance

2

Research Objectives

- How object-oriented (OO) software metrics can be used in iterative software processes to:
  - Predict source line changes within each class from one iteration to the next.
  - Reveal how OO design structures evolve in iterative processes and compare the evolution patterns with previous studies.
  - Validate software metrics including the design instability metric.

3

Research Hypotheses

- *Hypothesis 1:* Using OO metrics, we can predict source line changes in classes from one iteration to the next in the long-cycled framework iterative process.
- *Hypothesis 2*: Using OO metrics, we can predict source line changes in classes from one iteration to the next in the short-cycled Extreme Programming (XP) iterative process.

4

Research Hypotheses

- *Hypothesis 3*: Using OO metrics, we can predict maintenance effort, measured in man-hour, in classes from one iteration to the next in the short-cycled XP iterative process.
- *Hypothesis 4*: Using OO metrics, we can predict refactoring effort, measured in man-hour, in classes from one iteration to the next in the short-cycled XP iterative process.

5

Research Hypotheses

- *Hypothesis 5*: The design evolution should exhibit similar patterns in the short- and long-cycled iterative processes as it did in previous non-iterative studies.
- *Hypothesis 6*: System Design Instability (SDI) can indicate project progress in both the short- and the long-cycled iterative process as it did in the previous study for the non-iterative process.

6

Research Hypotheses

- *Hypothesis 7:* Class size has a strong impact on predicting design changes in the long-cycled framework iterative process.
- *Hypothesis 8:* Class size has a strong impact on predicting design changes in the short-cycled XP iterative process.

7

Introduction

- Metrics
- Software metrics
- Object-oriented software metrics
- Software processes

8

Metrics Suites

- **Chidamber and Kemerer**
  - WMC, DIT, NOC, CBO, RFC and LCOM
- **Li**
  - NAC, NDC, NLM, CMC, CTA and CTM

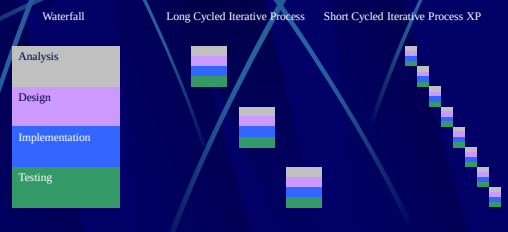
9

Previous Software Metrics Studies

- Li and Henry (J. of systems and software 1993).
- Basili *et al.* (TSE 1996).
- Briand *et al.* (TSE 1999).
- Fenton and Neil (TSE 1999).
- Fenton and Ohlsson (TSE 2000).
- El Emam *et al.* (TSE 2001)
- Briand and Wüst (TSE 2001)

10

Software Processes



Kent Beck 1999

11

Framework Iterative Process

- Framework system:
  - Collection of classes.
  - Built into a cohesive inheritance hierarchy,
  - Reusable and semi-complete application,
  - Provide more comprehensive reuse than classes developed by individual programmers.
  - Shipped as components used to build applications.
- Examples: CORBA, MFC and JFC.

12

Extreme Programming (XP)

- A new software process.
- Convenient and effective for projects that have vague requirements or the requirements are likely to change during development .
- XP features: stories, pair programming, unit testing and continuous integration.
- XP activities: New design, Error fix and Refactoring

13

Research Data

- To answer hypotheses 1 and 7:

"Hypothesis 1: Using OO metrics, we can predict source line changes in classes from one iteration to the next in the long-cycled framework iterative process."

"Hypothesis 7: Class size has a strong impact on predicting design changes in the long-cycled framework iterative process."

  - Various releases of JDK (JDK1.0, JDK1.1, JDK1.2, JDK1.3 and JDK 1.4)
    - Long evolutionary history,
    - Widely used in industry,
    - JDK releases changed throughout their development process.
    - Open source,

14

Research Data

- To answer hypotheses 2, 3, 4, and 8:

"Hypothesis 2: Using OO metrics, we can predict source line changes in classes from one iteration to the next in the short-cycled (XP) iterative process."

"Hypothesis 3: Using OO metrics, we can predict maintenance effort, measured in man-hour, in classes from one iteration to the next in the short-cycled XP iterative process."

"Hypothesis 4: Using OO metrics, we can predict refactoring effort, measured in man-hour, in classes from one iteration to the next in the short-cycled XP iterative process."

"Hypothesis 8: Class size has a strong impact on predicting design changes in the short-cycled XP iterative process."

  - Two OO systems:
    - Developed using Java, 2-year period of time, Data collected in daily basis.

15

Research Data

- To answer hypotheses 5 and 6

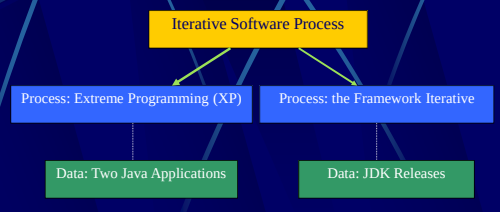
"Hypothesis 5: The design evolution should exhibit similar patterns in the short- and long-cycled iterative processes as it did in previous non-iterative studies."

"Hypothesis 6: System Design Instability (SDI) metric can indicate project progress in both the short- and the long-cycled iterative process as they did in the previous study for the non-iterative process."

  - Combination of the data we plan to use to answer the other hypotheses.

16

Research Data



17

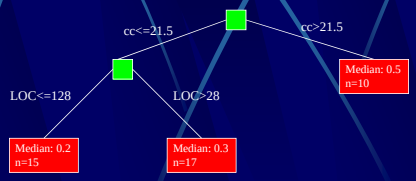
Research Approach

- Regression Trees:
  - Data mining technique.
  - Builds partition tree of the data set.
- Multiple Linear Regression (MLR):

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k + \varepsilon$$

18

Regression Trees



19

Research Significance

- Empirical validation of metrics is very important if the metrics are to be used.
- Validation of the metrics in OO iterative processes has never been done before
  - XP: short iterative cycle
  - Framework: long iterative cycle
- Try different prediction models
  - Multiple Linear Regression (MLR)
  - Regression Trees

20

Research Significance

- Examine the effect of class size on the prediction models.
- Validate the system design instability (SDI) metric in the two iterative processes.
- Reveal how OO design structures evolve in the two iterative processes and compare the evolution with previous results.

21

Summary

- Three objectives.
- Eight hypotheses.
- Approach to test the hypotheses.

22

Questions?

23

Prediction Example

Given the following data: number of lines deleted from JDK1.1-JDK1.2 in each class.

File Name	Lines Deleted	CTM	NLM	WMC	CTA	LCOM
Container.java	17	57	48	150	6	2117
Hashtable.java	23	12	19	53	1	113
InetAddress.java	23	14	10	24	2	136
Point.java	3	2	10	11	0	0
Runtime.java	31	10	17	22	0	253
String.java	16	22	48	126	0	573
Window.java	30	41	21	56	6	798
Boolean.java	3	1	6	7	1	9

24

Prediction Example

- The dependent variable:
  - Lines deleted.
- The independent variables are:
  - CTM
  - NLM
  - WMC
  - CTA
  - LCOM

25

Prediction Example

- Determine which independent variables are significant predictors of the dependent variable, and which independent variables can be eliminated.
  - Find the best subset.

26

Prediction Example

Best Subsets Regression: Lines Changed versus CTM, NLM, WMC, CTA, LCOM

Response is Lines Ch

Vars	R-Sq	R-Sq(adj)	C-p	S	C T M	N L M	W M C	C T A	L O M
1	14.5	0.3	5.2	10.791	X				
1	10.2	0.0	5.7	11.060				X	
2	44.1	21.7	4.0	9.5604	X				X
2	22.2	0.0	6.4	11.275	X	X			
3	54.3	20.0	4.9	9.6632	X	X			X
3	51.2	14.5	5.3	9.9907	X		X	X	
4	81.3	56.4	4.0	7.1358	X	X			X
4	75.1	42.0	4.7	8.2330	X		X	X	X
5	81.5	35.3	6.0	8.6959	X	X	X	X	X

27

Prediction Example

Regression Analysis: Lines Changed versus CTM, NLM, CTA, LCOM

The regression equation is

Lines Changed = 20.3 + 3.86 CTM - 1.53 NLM - 15.8 CTA - 0.0261 LCOM

Predictor	Coef	SE Coef	T	P
Constant	20.292	7.192	2.82	0.067
CTM	3.861	1.218	3.17	0.051
NLM	-1.5259	0.6910	-2.21	0.114
CTA	-15.796	6.474	-2.44	0.093
LCOM	-0.02611	0.01254	-2.08	0.129

S = 7.136 R-Sq = 81.3% R-Sq(adj) = 56.4%

Analysis of Variance

Source	DF	SS	MS	F	P
Regression	4	664.74	166.18	3.26	0.179
Residual Error	3	152.76	50.92		
Total	7	817.50			

Source	DF	Seq SS
CTM	1	118.80
NLM	1	27.76
CTA	1	297.42
LCOM	1	220.75

28